Domain Adaptation for Real-World Single View 3D Reconstruction

Brandon Leung
UC San Diego
b7leung@ucsd.edu

Siddharth Singh
UC San Diego
sisingh@eng.ucsd.edu

Arik Horodniceanu
UC San Diego
ahorodni@eng.ucsd.edu

Abstract

Deep learning-based object reconstruction algorithms have shown remarkable improvements over classical methods. However, supervised learning based methods perform poorly when the training data and the test data have different distributions. Indeed, most current works perform satisfactorily on the synthetic ShapeNet dataset, but dramatically fail in when presented with real world images. To address this issue, unsupervised domain adaptation can be used transfer knowledge from the labeled synthetic source domain and learn a classifier for the unlabeled real target domain.To tackle this challenge of single view 3D reconstruction in the real domain, we experiment with a variety of domain adaptation techniques inspired by the maximum mean discrepancy (MMD) loss, Deep CORAL, and the domain adversarial neural network (DANN). From these findings, we additionally propose a novel architecture which takes advantage of the fact that in this setting, target domain data is unsupervised with regards to the 3D model but supervised for class labels. We base our framework off a recent network called pix2vox. Results are performed with ShapeNet as the source domain and domains within the Object Dataset Domain Suite (ODDS) dataset as the target, which is a real world multiview, multidomain image dataset. The domains in ODDS vary in difficulty, allowing us to assess notions of domain gap size. Our results are the first in the multiview reconstruction literature using this dataset.

1. Introduction

Humans are able to understand the visual world in 3D. This helps us manipulate objects and intuitively make sense of the world. Crucially, we don't need to observe everyday objects from all viewpoints in order to have this ability. Usually only one views will suffice for us to have a strong understanding of the 3D object – for example, we can imagine the object from different novel viewpoints and "mentally rotate" them in our heads. In computer vision, this task is called *single view 3D reconstruction*: given a single 2D input image, we wish to output 3D object models (eg in the form of voxels, point clouds, or triangular meshes). Note that this is different from the classical problem of structure from motion, which requires a dense, full coverage of all viewpoints [19]. The single view case is difficult because in general, it is highly unconstrained. For instance, given the view of picture of a car from the front, no algorithm or person can produce an image of the car from the back with 100 percent certainty. However, there are certain clues which can lead to a reasonable result since we know that the world follows certain geometric patterns and rules. Additionally, since we have seen many pictures of cars in the past, we can incorporate our prior knowledge specific to the car class.

Regardless, nearly all of them use a Synthetic dataset of mesh models called ShapeNet [5] for both training and testing. This is because it provides many 2D rendered images of objects, as well as their ground truth 3D representations. However, because they are all CAD models with little detail, no texture, and no background, models trained on ShapeNet do not perform well when presented images in the real world due to a domain gap between synthetic and real. To be practical, this model should be able to work with images in the real world, with complicated backgrounds. Therefore, techniques from the domain adaptation literature can be applied. This would allow us to transfer knowledge from a synthetic source domain with 3D ground truth to an real target domain without 3D ground truth.

In this paper, we present results which extend the work of a current state-of-the-art, synthetic-based, single view voxel reconstruction method called pix2vox [28], so that it can be applied in the real world. First, we provide a summary of the related literature in Section 2. Then, our rationale for choosing this framework to work on is discussed in Section 3. In particular, we utilize several domain adaptation methods based on the maximum mean discrepancy (MMD) loss, Deep CORAL, and the domain adversarial neural network (DANN). We also propose a novel architecture which takes advantage of the fact that in this setting, target domain data is unsupervised with regards to the 3D model but supervised for class labels. Then, we share our results in Section 4. We demonstrate that as is, pix2vox fails with real-world

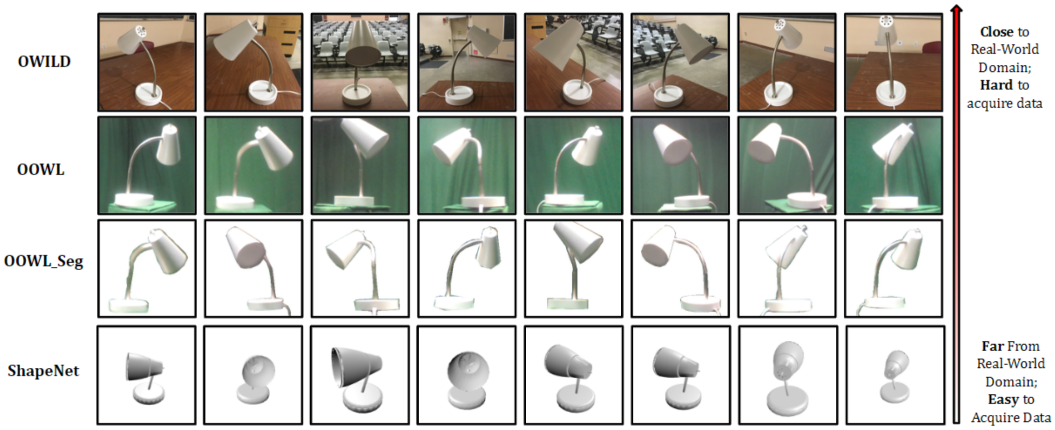
OWILD
OOWL
OOWL_Seg
ShapeNet
Close to Real-World Domain; Hard to acquire data
Far From Real-World Domain; Easy to Acquire Data

Figure 1. An example object (lamp) from datasets used for our project. OWILD, OOWL, and OOWLSeg are part of the ODDS dataset, and are real. Meanwhile, ShapeNet is synthetic.

images, evaluate the domain adaptation methods, and verify the usefulness incorporating class labels.

2. Related Work

2.1. 3D Single View Object Reconstruction

Single view 3D reconstruction has several immediate applications. For example, it would enable objects to be reconstructed as a 3D model and placed into an augmented (AR) or virtual (VR) environment, so that the user could manipulate them in that environment. Another use case is for robotic grasping – if an object such as a cup could be scanned, it would provide valuable information for a robot trying to pick it up by its handle. As a result, several methods have been proposed in the computer vision literature [28, 16, 6, 20, 25, 9]. They vary in the type of 3D representation used and each have their own trade-offs. For example, voxels are easily adapted to Convolutional Neural Networks (CNNs) but are spatially inefficient; triangular meshes are efficient but suffer from irregularities; point clouds are simple but lack explicit structural information. However, most of the these methods use ShapeNet for training and testing, and do not incorporate techniques used in the domain adaptation literature to allow for real-world viability. [27] uses Graph Convolutional Neural Net (GCN) to deform a mesh of ellipsoid to obtain an output mesh. The results, however, are not very accurate and it fails to keep the genus of the ground truth. [16] uses 2D convolutional network to generate dense point clouds that shapes the surface of 3D objects in an undiscretized 3D space. The method predicts accurate shapes with higher point density but is problematic when objects contain very thin structures. [6] proposed a novel architecture that unifies single and multi-view 3D reconstruction into a single framework. The method uses deep convolutional neural networks (3D Recurrent Reconstruction Neural Network) to learn a mapping from observations to their underlying 3D shapes of objects from a large collection of training data. It incrementally improves its reconstructions as it sees more views of an object but is unable to reconstruct many details and struggles with objects having high texture levels.

2.2. Unsupervised Domain Adaptation

Classical approaches to unsupervised domain adaptation usually consist of matching the feature distribution between the source and target domain. Generally these methods can be categorized as either sample re-weighting (eg. [13], [15]) or feature space transformations (eg. [2], [23]). Convolutional neural networks are also used for this purpose, because of their ability to learn powerful features. These methods, in general, are trained to minimize a classification loss while maximizing domain confusion. The classification loss is usually computed using a fully-connected or convolutional neural network trained on the labeled data. The domain confusion is usually achieved either by using a discrepancy loss, which reduces the shift between the two domains such as in ([17], [18], [4]) or via an adverserial loss which encourages a common feature space with respect to a discriminator loss, such as in [10], [4], [1]. [24] achieves the domain-confusion by aligning the second-order statistics of the learned feature representations. In [26] a domain-confusion loss based on MMD [12] is applied to the final layer representation of a network. [17] uses a sum of multiple MMDs between several layers and [18] continues this line of work by using the joint distribution discrepancy over deep features, instead of their sum.

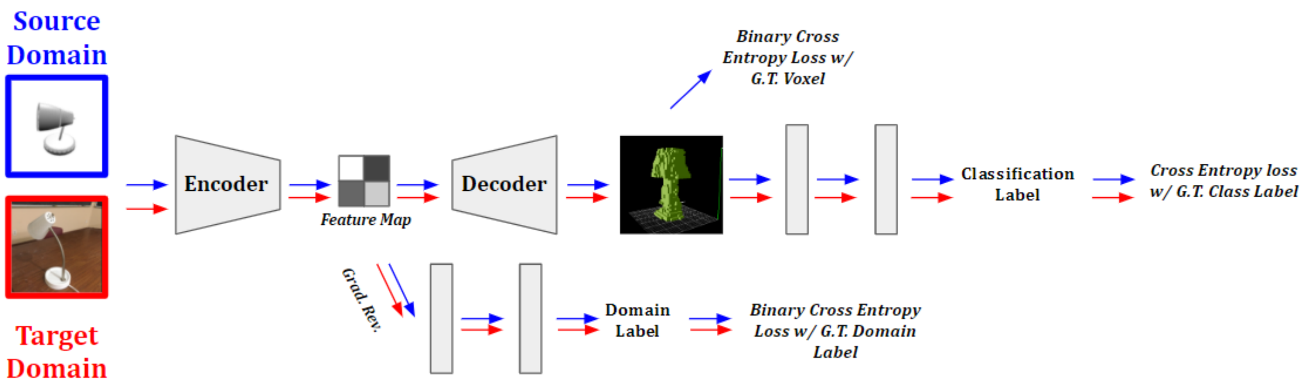
Source Domain
Target Domain
Encoder
Feature Map
Decoder
Binary Cross Entropy Loss w/ G.T. Voxel
Classification Label
Cross Entropy loss w/ G.T. Class Label
Grad. Rev.
Domain Label
Binary Cross Entropy Loss w/ G.T. Domain Label

Figure 2. The proposed architecture, which is based off the pix2vox framework. We provide two additions: 1) incorporating domain adaptation in the style of DANN through a domain classifier and a gradient reversal layer, and 2) since we have classification labels for both the source and target domain, we also classify the produced voxel.

3. Methods

3.1. 3D Reconstruction Backbone Architecture

The backbone for single view 3D reconstruction that was chosen for this project is the pix2vox architecture [28]. It is a convolutional neural network which encodes input images into a latent feature map, which is then decoded into a $32 \times 32 \times 32$ dimensional voxel. The encoder uses convolutional layers while the decoder uses 3D transpose convolutional layers. An additional refiner CNN which is based on the U-Net is also employed to increase performance [21]. Standard techniques are used throughout the network, including batch normalization [14], ReLU and ELU layers, and ImageNet [8] weights pretrained on VGG [22]. This architecture was chosen because it is much more efficient than other competing methods such as PSGN [9], OGN [25], and 3D-R2N2 [6], while being comparable or better in terms of performance. Due to limited computational resources, this was a critical factor in our decision. Note that in the original paper, pix2vox utilizes the ShapeNet dataset.

3.2. Maximum Mean Discrepancy (MMD)

Unsupervised domain adaptation is quite challenging since we do not have labeled information for the target domain. Some approaches to the problem are to try to bound the target error by the source error plus a discrepancy metric between the source and the target. The Maximum Mean Discrepancy (MMD) is a measure of the difference between two probability distributions from their samples. It is an effective criterion that compares distributions without initially estimating their density functions. Given two probability distributions p and q on $\mathcal{X}$, MMD is defined as

$$\mathcal{MMD}(\mathcal{F}, p, q) = sup_{f \in \mathcal{F}}(E_{x \sim p}[f(x)] - E_{y \sim q}[f(y)]) \tag{1}$$

where $\mathcal{F}$ is a class of functions $f : \mathcal{X} - \mathcal{R}$. By defining $\mathcal{F}$ as the set of functions of the unit ball in a universal Reproducing Kernel Hilbert Space (RKHS), denoted by $\mathcal{H}$, it was shown that $\mathcal{MMD}(\mathcal{F}, p, q) = 0$ will detect any discrepancy between p and q [3].

Let $x_s^{(i)}{}_{i=1,\ldots,n_s}$ and $x_t^{(j)}{}_{i=1,\ldots,n_t}$ be data vectors drawn from distributions $\mathcal{D}_s$ and $\mathcal{D}_t$ on the data space $\mathcal{X}$, respectively. Since f is in the unit ball in a universal RKHS, we can rewrite the empirical estimate of MMD as

$$\mathcal{MMD}_e(x_s, x_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_s^{(i)}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_t^{(j)})) \right\|_{\mathcal{H}} \tag{2}$$

where $\phi(\cdot) : \mathcal{X} \to \mathcal{H}$ is referred to as the feature space map.

3.3. Deep CORAL

Another approach to domain adaptation is by aligning the statistics of the source and target domains. CORAL [23] does this by using a linear transformation to align the covariances (second order statistic) of the domains. Assuming we have a labeled source domain $\mathcal{D}_s = \{x_s^{(i)}, y_s^{(i)}\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}_t = \{x_t^{(j)}\}_{j=1}^{n_t}$ where each sample is a d dimensional vector, the CORAL loss is defined as:

$$\ell_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|_F^2 \tag{3}$$

Where $\|\cdot\|_F$ is the Frobenius norm and $C_S, C_T \in \mathbb{R}^{d \times d}$ are the feature covariance matrices for the the source and target data, respectively. These matrices are given by:

$$C_S = \frac{1}{n_s - 1}(D_S^T D_S - \frac{1}{n_s}(\mathbf{1}^T D_S)^T(\mathbf{1}^T D_S)) \tag{4}$$

$$C_T = \frac{1}{n_t - 1}(D_T^T D_T - \frac{1}{n_t}(\mathbf{1}^T D_T)^T(\mathbf{1}^T D_T)) \tag{5}$$

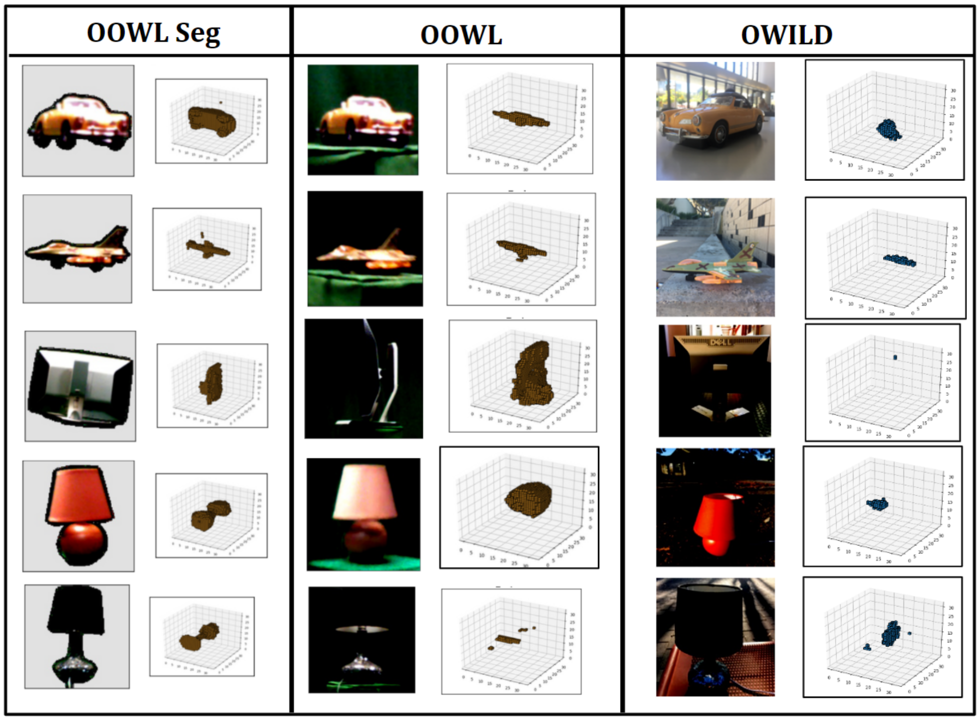

Figure 3. Reconstruction results of applying our proposed architecture on the three domains in the ODDS dataset; ShapeNet is used as the source domain, and each are trained on their respective target domains.

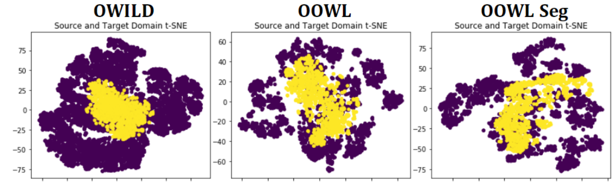

Figure 4. Learned feature map embeddings visualized with t-SNE, using our proposed model on the three target domains in the ODDS dataset. Purple denotes the source domain (ShapeNet), yellow denotes the target domain (OWILD, OOWL, and OOWL Seg).

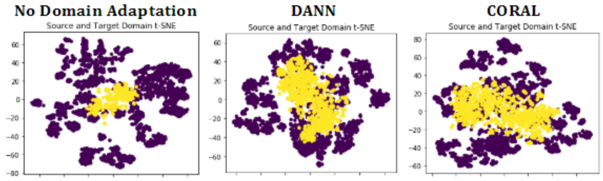

Figure 5. Learned feature map embeddings visualized with t-SNE. Purple denotes the source domain (ShapeNet), yellow denotes the target domain (OOWL). On the left we show the result when domain adaptation is not used. On the center and right, we show results achieved by applying domain adaptation (DANN or CORAL) to the vanilla pix2vox model.

Where **1** is a d dimensional vector containing all 1 and the matrices $D_S \in \mathbb{R}^{n_s \times d}, D_T \in \mathbb{R}^{n_t \times d}$ are the data matrices containing the source and target data, respectively.

3.4. Domain Adversarial Neural Network (DANN)

DANN [11] focuses on combining domain adaptation and deep feature learning under one training process. It embeds the domain adaptation method into the process of learning representation to obtain features which are discriminative and domain invariant. This is achieved by jointly optimizing the underlying features as well as two discriminative classifiers operating on these features, the *label predictor and domain classifier*. The *label predictor* predicts class labels and is used both during training and at test time. *domain classifier* discriminates between the source and the target domains during training. The model works to *minimize* the loss of the label classifier and *maximize* the domain classifier loss adversarially, thereby encouraging domain-invariant features.

3.5. Voxel Classification Architecture

The domain adaptation techniques discussed above can be readily applied to the pix2vox architecture. However, their success may be limited, since the gap between synthetic and real domains is large. To help with this process, in addition to using domain adaptation techniques, we perform classification of the output voxel by vectorizing it, and using several fully connected layers of size 100 and 20, with ReLU activations. This is possible since we have the ground truth class labels for both the source and target domain. This proposed architecture can be seen in Figure 2; all losses are trained end-to-end. We utilize the standard cross entropy loss. This idea is inspired by the fact that in general, the output voxels should resemble their respective classes. We found that this additional source of supervision, which applies to both the source and target domain, is highly beneficial; further details can be found in Section 4.

4. Experiments

4.1. Relevant Datasets

As mentioned previously, ShapeNet is a synthetic dataset which provides many 2D rendered images of objects, as well as their ground truth 3D mesh representations (which can be converted into voxels). As shown in Figure 1, they are CAD models with little detail, no texture, and no background. While the original ShapeNet has 270 classes, pix2vox uses a subset of 13 classes.

In addition, we utilize the ODDS dataset, which is a real, multiview, class-organized image dataset with multiple domains. It contains 25 classes, 6 of which overlap with the ShapeNet classes used to train the original pix2vox model. The overlapping classes are airplanes, cars, monitors, lamps, telephones, and boats, so these are the classes we use when comparing results between datasets. Each ODDS class contains 20 object instances (for example, the monitor class has 20 different types of monitors); each object instance has 8 images of it taken at 45 degree increments. Please see Figure 1 for some example images. There are 3 domains in the ODDS dataset that we work with. First, OWILD contains images of objects in various real-world locations, and pictures are taken with a smartphone. Second, OOWL is taken with a drone inside a lab setting. As a result, it contains several domain peculiarities such as as camera blur and a lower camera resolution. Finally, OOWLSeg is a segmented version of OOWL. Note that this data captures the real-world input statistics that we would like to work with: they're real objects, taken with smartphone cameras in various real-world locations. However, they also represent a trade-off between ease of collection and realism – this is shown through the arrow in Figure 1 on the right.

4.2. Evaluation Metrics

The standard evaluation for 3D reconstruction when using voxels is the intersection over union (IoU) score. Formally, it is:

$$IoU = \frac{\sum_{i,j,k} I(p_{(i,j,k)} > t) I(gt_{(i,j,k)})}{\sum_{i,j,k} I[I(p_{(i,j,k)} > t) + I(gt_{(i,j,k)})]} \qquad (6)$$

where $p_{(i,j,k)} \in [0,1]$ is the predicted occupancy probability at voxel location (i,j,k) and $gt_{(i,j,k)} \in \{0,1\}$ is the ground truth value at voxel location (i,j,k). Given a predicted reconstruction voxel and ground truth voxel, if $IoU = 1$, then they are the same voxel. If $IoU = 0$, then there is no intersection between the two voxels. Thus, a higher IoU score indicates a better reconstruction result. It is important to note that for the case of the ODDS dataset, we do not have ground truth (currently, we are not aware of any publicly available, sufficiently large multiview dataset with 3D ground truth). Therefore, for the scope of this paper, it is primarily used as a test dataset to judge qualitative reconstruction results. We cannot quantitatively evaluate metrics like IoU, due to the lack of 3D ground truth (it is unsupervised in this regard). However, we do try to utilize the ground truth class labels that come with OWILD as supervision.

4.3. Application of Domain Adaptation on a Vanilla Pix2Vox Model

First, we report the results of applying DANN and Deep CORAL domain adaptation to the vanilla pix2vox model for single view 3D reconstruction. Qualitative reconstruction results are shown in Figure 7, under "CORAL" and "DANN". We observed that reconstruction when not using any domain adaptation generally looks like random noise (these reconstruction results are omitted to save space). Also, in our experiments we found that MMD was not effective; therefore, MMD results have also been omitted. Regarding Deep CORAL and DANN, we can see that in general both help, though results are still far from perfect. Visually checking the reconstruction results, we found that DANN performed better than CORAL. We also embed the learned feature maps into 2D, using t-SNE as the dimensionality reduction algorithm. This is shown in Figure 5. We can see that the use of DANN and Deep CORAL both helps to make embedded features more domain invariant – the distributions of the source ShapeNet domain (purple) are more matched with the distributions of the target OOWL domain (yellow). We also note that the introduction of domain adaptation also negatively impacts the IoU on the source domain – this is shown in Table 1. Intuitively, this makes sense since the network is constrained to only output domain invariant latent representations. This makes training more difficult. In the future, we would like to look into this more and see if we can maintain IoU results on the source domain while performing domain adaptation.

4.4. Reconstruction with a Voxel Classification Loss

As mentioned above, we found that regarding the vanilla pix2vox model, DANN is helpful. However, results are still sub optimal. To address this, we utilize our proposed voxel classification network. Training is performed end-to-end,

Class	No DA	DANN	CORAL
Airplane	0.6842	0.6046	0.6377
Car	0.8548	0.8377	0.8485
Monitor	0.5373	0.4845	0.4968
Lamp	0.4430	0.4228	0.4322
Telephone	0.7764	0.7278	0.7555
Boat	0.5946	0.5852	0.5926
Overall	0.7110	0.6762	0.6925

Table 1. IoU results for a model trained on ShapeNet with and without OOWL domain adaptation (DANN, Deep CORAL). Results reported for $t = 0.4$.

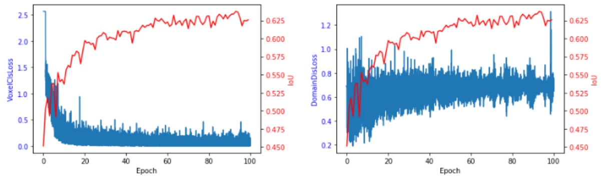

Figure 6. Training results on our architecture. On the left, the voxel classification loss (in blue) is plotted against the IoU. On the right, the domain discrepancy loss (in blue) is plotted against the IoU.

and we report training losses as a function of epoch in Figure 6. While the voxel classification loss does decrease, during training we found it difficult to reduce it past epoch 20 – it fluctuates beyond that point. In the future, we plan on trying to look into ways of address this. Meanwhile, the domain discrepancy loss is maintained at around 0.5. This is expected due to adversarial training induced by the gradient reversal layer.

Next, using this trained model we evaluate the differences between the domains in ODDS. This gives us a way to see how large the domain gap is between Shapenet and the target domains OOWL, OOWL Seg, and OWILD. We report t-SNE embeddings in Figure 4 and the reconstructions in Figure 3 for the three target domains. We can see that in general, it appears that OWILD is the most challenging dataset. We believe that this is because OWILD has complex backgrounds, which make it very far from the shapenet domain. On the other hand, we can see that OOWL seg performs quite well. We believe that the segmentation makes the images very similar to ShapeNet, which also do not have a background.

5. Conclusion and Future Work

In this paper, we have focused on the task of single view voxel reconstruction in the real world. To do this, we extended the pix2vox architecture using domain adaptation between the supervised synthetic ShapeNet dataset and the unsupervised, real ODDS dataset. However, we showed that simply applying domain adaptation is not enough; reconstruction results are only marginally better. Therefore, we proposed an architecture which also utilizes a voxel classification loss in addition to an adversarial loss, which led to better results.

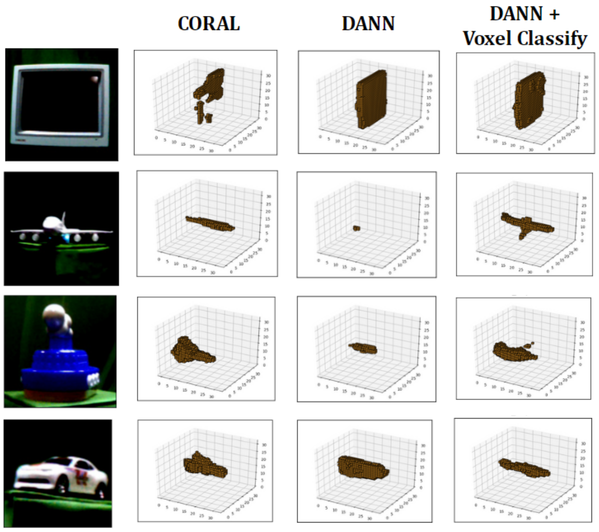

Figure 7. Reconstruction results when domain adaptation is used with OOWL as the target domain. We compare domain adaptation using Deep CORAL, DANN, and our proposed architecture (DANN + Voxel Classify).

There are several extensions that were not done due to time and computational constraints which we plan on exploring in the future. First, it would be interesting to see if our conclusions in the project hold for other architectures (eg mesh or point clouds). Second, no large dataset exists with real world ground truth 3D data. Perhaps the closest is Pix3D, but it only has 8 classes, and pictures are only from one angle. No class overlaps with Pix3D, OWILD, and ShapeNet. If such a dataset existed, it would be feasible to achieve quantitative, not just qualitative reconstruction results. Third, we plan on working towards improving results on OWILD through more experimentation. For example, we want to try data augmentation on ShapeNet by pasting background from the MIT Places datasets [29] and see if that improves results. We also want to further explore the domain gaps between the datasets through methods like domain bridges, which proposes domain adaptation gradually over several intermediate domains which increase in difficulty to the final target domain [7]. Finally, because ODDS is a multiview dataset, it would be natural to generalize results to the multiview reconstruction case.

Acknowledgements. The team would like to thank Professor Manmohan Chandraker and TA Yu-Ying Yeh for an informative and well taught class in CSE 291.

6. Team Member Responsibilities

Brandon Leung: Implemented and conducted all reconstruction and t-SNE experiments related to Deep CORAL, DANN, and the proposed architecture.

Siddharth Singh: Implemented and conducted all experiments related to MMD.

Arik Horodniceanu: Contributed to writing sections 1, 2, and 3 of this paper.

References

[1] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand. Domain-adversarial neural networks, 2014. 2

[2] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013. 2

[3] K. Borgwardt, A. Gretton, M. Rasch, H.-P. Kriegel, B. Schölkopf, and A. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics (Oxford, England)*, 22:e49–57, 08 2006. 3

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. *CoRR*, abs/1608.06019, 2016. 2

[5] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[6] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2, 3

[7] S. Dai, K. Sohn, Y.-H. Tsai, L. Carin, and M. Chandraker. Adaptation across extreme variations using unlabeled domain bridges. *arXiv preprint arXiv:1906.02238*, 2019. 6

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[9] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2, 3

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks, 2015. 2

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. *Domain-Adversarial Training of Neural Networks*, pages 189–209. Springer International Publishing, Cham, 2017. 4

[12] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel method for the two-sample problem, 2008. 2

[13] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, pages 601–608, Cambridge, MA, USA, Sept. 2007. Max-Planck-Gesellschaft, MIT Press. 2

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[15] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 2

[16] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[17] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks, 2015. 2

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[19] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 1

[20] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1936–1944, 2018. 2

[21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[23] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation, 2015. 2, 3

[24] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation, 2016. 2

[25] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2, 3

[26] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance, 2014. 2

[27] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y. Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. *CoRR*, abs/1804.01654, 2018. 2

[28] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2690–2698, 2019. 1, 2, 3

[29] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 6